Controllo MSComm (comunicazioni)

Il controllo **MSComm** (comunicazioni) fornisce all'applicazione funzioni per le comunicazioni seriali, consentendo la trasmissione e la ricezione di dati tramite una porta seriale.

Sintassi

MSComm

Osservazioni

Il controllo **MSComm** consente di gestire le comunicazioni nei due modi seguenti:

- La gestione delle comunicazioni tramite eventi è molto efficace per il controllo delle interazioni tra porte seriali. In molte situazioni può essere utile ricevere un segnale nel momento in cui viene generato un determinato evento, quale l'arrivo di un carattere o una modifica sulle linee CD (Carrier Detect) o RTS (Request To Send). In tali casi, l'evento OnComm del controllo **MSComm** consente di intercettare e gestire gli eventi di comunicazione. L'evento OnComm consente inoltre di individuare e gestire gli errori di comunicazione. Per un elenco di tutti gli eventi e gli errori di comunicazione, vedere la proprietà **CommEvent**.

- Il polling di eventi ed errori può essere inoltre eseguito controllando il valore della proprietà **CommEvent** dopo l'esecuzione di ogni funzione critica del programma. Questo metodo è consigliato nel caso di applicazioni autonome e di dimensioni ridotte. Se ,ad esempio, si sta scrivendo un semplice programma di composizione di numeri telefonici, potrebbe non essere necessario generare un evento dopo la ricezione di ciascun carattere, in quanto il solo carattere atteso è il segnale OK di risposta del modem.

Ogni controllo **MSComm** utilizzato corrisponde a una porta seriale. Per accedere a più porte seriali con l'applicazione, sarà necessario utilizzare più controlli **MSComm** . L'indirizzo di porta e di interrupt può essere modificato tramite il Pannello di controllo di Windows.

Per il controllo **MSComm** sono disponibili diverse proprietà importanti. Di seguito sono riportate le proprietà fondamentali.

Proprietà	Descrizione
CommPort	Imposta e restituisce il numero della porta di comunicazioni.
Settings	Imposta e restituisce in forma di stringa la velocità, la parità, i bit di dati e i bit di stop.
PortOpen	Imposta e restituisce lo stato di una porta di comunicazioni, oltre ad attivare e disattivare la porta.
Input	Restituisce e rimuove caratteri dal buffer di ricezione.
Output	Scrive una stringa di caratteri nel buffer di trasmissione.

Proprietà CommEvent

Restituisce l'evento o l'errore di comunicazione più recente. La proprietà non è disponibile in fase di progettazione ed è di sola lettura in fase di esecuzione.

Sintassi

oggetto.**CommEvent**

La sintassi della proprietà **CommEvent** è composta dalle seguenti parti:

Parte	Descrizione
oggetto	Espressione oggetto che definisce un oggetto dell'elenco "Si applica a".

Osservazioni

L'evento OnComm viene generato in corrispondenza di ogni evento o errore di comunicazione e il codice corrispondente viene memorizzato nella proprietà **CommEvent**. Per determinare l'errore o l'evento che ha generato l'evento OnComm, sarà pertanto necessario fare riferimento alla proprietà **CommEvent**.

Il codice restituito dalla proprietà **CommEvent** può essere uno dei valori indicati nella tabella seguente per gli errori o gli eventi di comunicazione. Le costanti corrispondenti sono inoltre disponibili nella libreria degli oggetti di questo controllo.

Le possibili impostazioni degli errori di comunicazione sono:

Costante	Valore	Descrizione
comEventBreak	1001	Ricezione di un segnale di interruzione.
comEventFrame	1004	Errore di frame. L'hardware ha individuato un errore di frame.
comEventOverrun	1006	Overrun della porta. L'hardware non ha letto un carattere prima dell'arrivo del successivo e il carattere è andato perduto.
comEventRxOver	1008	Overflow del buffer di ricezione. Spazio esaurito nel buffer di ricezione.
comEventRxParity	1009	Errore di parità. È stato rilevato un errore di parità.
comEventTxFull	1010	Buffer di trasmissione pieno. Spazio esaurito nel buffer di trasmissione durante il tentativo di inserimento di un carattere.
comEventDCB	1011	Errore imprevisto durante il recupero del DCB (Device Control Block) della porta.

Le possibili impostazioni degli eventi di comunicazione sono:

Costante	Valore	Descrizione
comEvSend	1	Nel buffer di trasmissione è presente un numero di caratteri inferiore a quello definito dalla proprietà **Sthreshold**.
comEvReceive	2	Numero di caratteri **Rthreshold** ricevuti. Questo evento viene generato fino a quando non si utilizza la proprietà **Input** per rimuovere i dati dal buffer di ricezione.
comEvCTS	3	Modifica nella linea CTS (Clear To Send).
comEvDSR	4	Modifica nella linea DSR (Data Set Ready). Questo evento viene generato solo quando DSR cambia da 1 a 0.
comEvCD	5	Modifica nella linea CD (Carrier Detect).
comEvRing RSI	6	Individuato segnale telefonico. È possibile che alcuni UART (universal asynchronous receiver-transmitters) non supportino questo evento.
comEvEOF	7	Ricevuto carattere indicatore di fine file (carattere ASCII 26).

Tipo di dati

Integer

This document contains the following shortcuts:

Shortcut Text	Internet Address
libreria degli oggetti	javascript:alink_4.Click()

Proprietà CDHolding

Determina se è presente una portante, verificando lo stato della linea CD (Carrier Detect). Il segnale CD viene inviato da un modem al computer connesso per indicare che il modem è in linea. La proprietà non è disponibile in fase di progettazione ed è di sola lettura in fase di esecuzione.

Sintassi

oggetto.**CDHolding**

La sintassi della proprietà **CDHolding** è composta dalle seguenti parti:

Parte	Descrizione
oggetto	Espressione oggetto che definisce un oggetto dell'elenco "Si applica a".

Impostazioni

Le possibili impostazioni della proprietà **CDHolding** sono:

Impostazione	Descrizione
True	La linea CD è attiva
False	La linea CD non è attiva

Osservazioni

> **Nota** Il rilevamento della perdita della portante è particolarmente importante nel caso di un'applicazione host, quale una BBS, in quanto il chiamante può interrompere la comunicazione (perdere la portante) in qualsiasi momento.

La condizione CD è anche nota come RLSD (Receive Line Signal Detect).

Tipo di dati

Boolean

Proprietà CTSHolding

Determina se è possibile inviare i dati, verificando lo stato della linea CTS (Clear To Send). Il segnale CTS viene, in genere, inviato da un modem al computer connesso, per indicare che è possibile procedere con la trasmissione. La proprietà non è disponibile in fase di progettazione ed è di sola lettura in fase di esecuzione.

Sintassi

oggetto.**CTSHolding**

La sintassi della proprietà **CTSHolding** è composta dalle seguenti parti:

Parte	Descrizione
oggetto	Espressione oggetto che definisce un oggetto dell'elenco "Si applica a".

Le possibili impostazioni della proprietà **CTSHolding** per il controllo **MSComm** sono:

Impostazione	Descrizione
True	Linea CTS attiva.
False	Linea CTS non attiva.

Osservazioni

La linea CTS viene utilizzata per la sincronia hardware RTS/CTS (Request To Send/Clear To Send). La proprietà **CTSHolding** consente di eseguire manualmente il polling della linea CTS per determinarne lo stato.

Ulteriori informazioni Per informazioni sui protocolli di sincronia, vedere la proprietà **Handshaking**.

Tipo di dati

Boolean

Proprietà DSRHolding

Determina lo stato di una linea DSR (Data Set Ready). Un modem, in genere, invia il segnale DSR al computer al quale è connesso per indicare che è pronto per l'esecuzione. La proprietà non è disponibile in fase di progettazione ed è di sola lettura in fase di esecuzione.

Sintassi

oggetto.**DSRHolding**

Il segnaposto *oggetto* rappresenta un'espressione oggetto che definisce un oggetto dell'elenco "Si applica a".

La proprietà **DSRHolding** restituisce i seguenti valori:

Valore	Descrizione
True	La linea DSR è attiva
False	La linea DSR non è attiva

Osservazioni

Questa proprietà è utile per la scrittura di una routine di sincronia Data Set Ready/Data Terminal Ready per un computer DTE (Data Terminal Equipment).

Tipo di dati

Boolean

Evento OnComm

L'evento OnComm viene generato a ogni modifica del valore della proprietà **CommEvent**, per indicare che è stato generato un evento di comunicazione oppure un errore.

Sintassi

Private Sub *oggetto*_**OnComm ( )**

La sintassi dell'evento **OnComm** è composta dalle seguenti parti:

Parte	Descrizione
oggetto	Espressione oggetto che definisce un oggetto dell'elenco "Si applica a".

Osservazioni

La proprietà **CommEvent** contiene il codice numerico dell'errore o dell'evento che ha generato l'evento **OnComm**. Si noti che l'impostazione delle proprietà **RThreshold** o **SThreshold** su 0 disattiva rispettivamente l'individuazione degli eventi **comEvReceive** e **comEvSend**.

Controllo TextBox (casella di testo)

Il controllo **TextBox** (casella di testo), a volte definito campo di modifica o controllo di modifica, consente di visualizzare le informazioni immesse in fase di progettazione, immesse dall'utente o assegnate al controllo nel codice in fase di esecuzione.

Sintassi

TextBox

Osservazioni

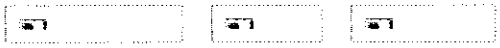
Per visualizzare più righe di testo in un controllo **TextBox**, impostare la proprietà **MultiLine** su **True**. Se un controllo **TextBox** a più righe non include una barra orizzontale, al testo viene automaticamente applicato il ritorno a capo automatico anche quando il controllo viene ridimensionato. Per personalizzare la combinazione delle barre di scorrimento in un controllo **TextBox**, impostare la proprietà **ScrollBars**.

Le barre di scorrimento vengono visualizzate accanto alla casella di testo quando la proprietà **MultiLine** di quest'ultima è impostata su **True** e la proprietà **ScrollBars** è impostata su un valore diverso da **None** (0).

Se si imposta la proprietà **MultiLine** su **True**, è possibile utilizzare la proprietà **Alignment** per impostare l'allineamento del testo all'interno del controllo **TextBox**. Per impostazione predefinita il testo viene allineato a sinistra. Se la proprietà **MultiLine** è impostata su **False**, l'impostazione di **Alignment** non avrà alcun effetto.

Il controllo **TextBox** può fungere da collegamento di destinazione in una conversazione DDE.

Proprietà Text



- Con il controllo **ComboBox** (proprietà **Style** impostata su 0 [elenco a discesa combinato] o su 1 [casella combinata semplice]) e il controllo **TextBox**, restituisce o imposta il testo contenuto nell'area di modifica.

- Con il controllo **ComboBox** (proprietà **Style** impostata su 2 [elenco a discesa]) e il controllo **ListBox**, restituisce l'elemento selezionato nella casella di riepilogo. Il valore restituito è sempre equivalente al valore restituito dall'espressione `List(ListIndex)`. Di sola lettura sia in fase di progettazione che in fase di esecuzione.

Sintassi

oggetto.**Text** [= *stringa*]

La sintassi della proprietà **Text** è composta dalle seguenti parti:

Parte	Descrizione
oggetto	Espressione oggetto che definisce un oggetto dell'elenco "Si applica a".
stringa	Espressione stringa che specifica il testo.

Osservazioni

Solo in fase di progettazione, le impostazioni predefinite della proprietà **Text** sono:

- Per i controlli **ComboBox** e **TextBox**, la proprietà **Name** del controllo.

- Per il controllo **ListBox**, una stringa vuota ("").

In un controllo **ComboBox** con la proprietà **Style** impostata su 0 (elenco a discesa combinato) o su 1 (casella combinata semplice) e in un controllo **TextBox**, questa proprietà risulta utile per leggere la stringa contenuta nell'area di modifica del controllo. In un controllo **ComboBox** o **ListBox** con la proprietà **Style** impostata su 2 (elenco a discesa), è possibile utilizzare la proprietà **Text** per determinare l'elemento selezionato.

L'impostazione di **Text** per un controllo **TextBox** può contenere al massimo 2048 caratteri, a meno che la proprietà **MultiLine** non sia **True**. In quest'ultimo caso il limite massimo è di circa 32K.

Proprietà DataChanged

Restituisce o imposta un valore che indica che i dati nel controllo associato sono stati modificati con processi diversi dal recupero dei dati dal record corrente. Non disponibile in fase di progettazione.

Sintassi

oggetto.**DataChanged** [= *valore*]

La sintassi della proprietà **DataChanged** è composta dalle seguenti parti:

Parte	Descrizione
oggetto	Espressione oggetto che definisce un oggetto dell'elenco "Si applica a".
valore	Espressione booleana che indica se i dati sono stati modificati come descritto nella sezione "Impostazioni".

Impostazioni

Le possibili impostazioni di *valore* sono:

Impostazione	Descrizione
True	I dati inclusi nel controllo sono diversi da quelli del record corrente.
False	(Predefinita) I dati eventualmente inclusi nel controllo sono gli stessi dati inclusi nel record corrente.

Osservazioni

Quando un controllo che gestisce dati si sposta da un record a un altro, i dati vengono passati dai campi del record corrente a controlli associati al campo specifico o all'intero record. Quando i dati vengono visualizzati nei controlli associati, la proprietà **DataChanged** è impostata su **False**. Se l'utente o una qualsiasi operazione modifica il valore nel controllo associato, la proprietà **DataChanged** verrà impostata su **True**. Il semplice spostamento a un altro record non ha effetto sulla proprietà **DataChanged**.

Quando il controllo che gestisce dati inizia lo spostamento a un altro record, viene generato l'evento Validate. Se la proprietà **DataChanged** di un qualsiasi controllo associato è impostata su **True**, il controllo richiamerà automaticamente i metodi **Edit** e **Update** per inviare le modifiche al database.

Se non si desidera salvare le modifiche da un controllo associato al database, è possibile impostare la proprietà **DataChanged** su **False** nell'evento Validate.

Verificare il valore della proprietà **DataChanged** nel codice per assicurarsi che l'evento Change di un controllo non comporti eventi a catena. Il suggerimento vale per i controlli

associati e non associati.

Tipo di dati

Integer (Boolean)

scorrimento in un controllo **TextBox**. La routine può inoltre essere utilizzata per visualizzare i dati e le formule in una determinata area di lavoro e i risultati in un'area diversa.

Le routine di eventi Change risultano utili per eseguire l'aggiornamento delle proprietà nei controlli del file system, ovvero **DirListBox**, **DriveListBox** e **FileListBox**. È possibile, ad esempio, aggiornare l'impostazione della proprietà **Path** per il controllo **DirListBox** dopo aver modificato l'impostazione della proprietà **Drive** del controllo **DriveListBox**.

Nota Una routine dell'evento Change può a volte causare un evento a catena. Ciò si verifica quando l'evento Change del controllo altera il contenuto del controllo impostando nel codice una proprietà che determina il valore del controllo, come ad esempio la proprietà **Text** per il controllo **TextBox**. Per impedire che venga generato un evento a catena:

- Se possibile, evitare di scrivere routine di eventi Change per controlli che modificano il contenuto del controllo. Se è necessario scriverla, impostare un flag che impedisca ulteriori modifiche mentre è in corso la modifica corrente.
- Evitare di creare due o più controlli con routine di eventi Change che interferiscono l'uno con l'altro, ad esempio due controlli **TextBox** che si aggiornano a vicenda durante l'esecuzione dei relativi eventi Change.
- Evitare di utilizzare una funzione o istruzione **MsgBox** in questo evento per i controlli **HScrollBar** e **VScrollBar**.

Evento Change

Indica che il contenuto di un controllo è stato modificato. Il momento e le modalità con cui questo evento viene generato dipendono dal controllo, come indicato di seguito:

- **ComboBox**: modifica il testo nella casella di testo del controllo. L'evento viene generato solo se la proprietà **Style** è impostata su 0 (casella combinata a discesa) o 1 (casella combinata semplice) e l'utente modifica il testo oppure viene modificata l'impostazione della proprietà **Text** nel codice.
- **DirListBox**: cambia la directory selezionata. L'evento viene generato quando si fa doppio clic su una directory diversa o quando viene modificata l'impostazione della proprietà **Path** nel codice.
- **DriveListBox**: cambia l'unità selezionata. L'evento viene generato quando viene selezionata un'unità diversa o modificata l'impostazione della proprietà **Drive** nel codice.
- **HScrollBar** e **VScrollBar** (barre di scorrimento orizzontale e verticale): spostano la casella di scorrimento nella barra di scorrimento. L'evento viene generato quando si esegue uno scorrimento o si modifica l'impostazione della proprietà **Value** nel codice.
- **Label**: modifica il contenuto di **Label**. L'evento viene generato quando viene eseguito un aggiornamento da un collegamento DDE o quando si modifica l'impostazione della proprietà **Caption** nel codice.
- **PictureBox**: modifica il contenuto di **PictureBox**. L'evento viene generato quando viene eseguito un aggiornamento da un collegamento DDE o quando si modifica l'impostazione della proprietà **Picture** nel codice.
- **TextBox**: modifica il contenuto della casella di testo. L'evento viene generato quando viene eseguito un aggiornamento mediante un collegamento DDE, quando l'utente modifica il testo o quando si modifica l'impostazione della proprietà **Text** nel codice.

Sintassi

Private Sub *oggetto*_**Change(**[*index* **As Integer**]**)**

La sintassi dell'evento Change è composta dalle seguenti parti:

Parte	Descrizione
oggetto	Espressione oggetto che definisce un oggetto dell'elenco "Si applica a".
index	Intero che identifica in modo univoco un controllo se incluso in una matrice di controlli.

Osservazioni

La routine dell'evento Change consente di sincronizzare o coordinare la visualizzazione dei dati in più controlli. È possibile, ad esempio, utilizzare la routine dell'evento Change di una barra di scorrimento per aggiornare l'impostazione della proprietà **Value** della barra di

Eventi KeyDown e KeyUp

Vengono generati quando si preme (KeyDown) o si rilascia (KeyUp) un tasto mentre un oggetto ha lo stato attivo. Se è necessario che vengano interpretati caratteri ANSI, utilizzare l'evento KeyPress.

Sintassi

Private Sub Form_KeyDown(*keycode* **As Integer**, *shift* **As Integer)**

Private Sub *oggetto***_KeyDown(**[*index* **As Integer**,]*keycode* **As Integer**, *shift* **As Integer)**

Private Sub Form_KeyUp(*keycode* **As Integer**, *shift* **As Integer)**

Private Sub *oggetto***_KeyUp(**[*index* **As Integer**,]*keycode* **As Integer**, *shift* **As Integer)**

La sintassi degli eventi KeyDown e KeyUp è composta dalle seguenti parti:

Parte	Descrizione
oggetto	Espressione oggetto che definisce un oggetto dell'elenco "Si applica a".
index	Intero che identifica in modo univoco un controllo se questo è incluso in una matrice di controlli.
keycode	Codice di un tasto, ad esempio **vbKeyF1** per il tasto F1 e **vbKeyHome** per il tasto HOME. Per specificare codici di tasto, utilizzare le costanti della libreria degli oggetti di Visual Basic (VB) nel Visualizzatore oggetti.
shift	Intero che indica lo stato dei tasti MAIUSC, CTRL e ALT nel momento in cui viene generato l'evento. L'argomento è un campo bit in cui i bit meno significativi corrispondono a MAIUSC (bit 0), CTRL (bit 1) e ALT (bit 2). Questi bit corrispondono rispettivamente ai valori 1, 2 e 4. È possibile impostare tutti i bit, uno solo o nessuno a seconda che si desideri indicare che tutti, uno solo o nessuno dei tasti viene premuto. Per indicare, ad esempio, che vengono premuti sia CTRL che ALT, si dovrà impostare l'argomento *shift* su 6.

Osservazioni

In entrambi gli eventi, l'oggetto che ha lo stato attivo riceve tutte le pressioni dei tasti. Un form può avere lo stato attivo solo quando i controlli non sono visibili e sono disattivati. Sebbene possano essere applicati alla maggior parte dei tasti, i due eventi KeyDown e KeyUp vengono in genere utilizzati per:

- Tasti di caratteri estesi, come ad esempio i tasti funzione.
- Tasti di spostamento.
- Combinazioni di tasti con modificatori di tastiera standard.

- Distinguere tra i tasti numerici del tastierino numerico e quelli della tastiera normale.

Utilizzare le routine di eventi KeyDown e KeyUp quando è necessario fornire una risposta sia alla pressione che al rilascio di un tasto.

KeyDown e KeyUp non vengono richiamati per:

- Il tasto INVIO se il form include un controllo **CommandButton** con la proprietà **Default** impostata su **True**.
- Il tasto ESC se il form include un controllo **CommandButton** con la proprietà **Cancel** impostata su **True**.
- Il tasto TAB.

Gli eventi KeyDown e KeyUp identificano i caratteri maiuscoli e minuscoli in base a due argomenti, ovvero l'argomento *keycode* che indica il tasto fisico e restituisce quindi "A" maiuscolo e "a" minuscolo come lo stesso tasto, e l'argomento *shift* che indica lo stato di *shift+tasto* e restituisce quindi "A" maiuscolo o "a" minuscolo.

Per verificare l'argomento *shift* è possibile dichiarare delle costanti che definiscano i bit dell'argomento. Alle costanti *shift* possono essere assegnati i seguenti valori:

Costante	Valore	Descrizione
vbShiftMask	1	Maschera di bit del tasto MAIUSC.
VbCtrlMask	2	Maschera di bit del tasto CTRL.
VbAltMask	4	Maschera di bit del tasto ALT.

Le costanti agiscono come maschere di bit che è possibile utilizzare per verificare qualsiasi combinazione di tasti.

Per provare una determinata condizione, è innanzitutto necessario assegnare ciascun risultato a una variabile intera temporanea e quindi confrontare *shift* con una maschera di bit. Utilizzare l'operatore **And** insieme all'argomento *shift* per verificare se la condizione è maggiore di 0, ovvero se il tasto modificatore è stato premuto, ad esempio:

```
ShiftDown = (Shift And vbShiftMask) > 0
```

In una routine è possibile provare qualsiasi combinazione di condizioni, ad esempio:

```
If ShiftDown And CtrlDown Then
```

Nota Se la proprietà **KeyPreview** è impostata su **True**, un form riceve questi eventi prima che li ricevano i controlli del form. Utilizzare questa proprietà per creare routine di gestione della tastiera globali.

Evento KeyPress

Viene generato quando si preme e si rilascia un tasto ANSI.

Sintassi

Private Sub Form_KeyPress(*keyascii* **As Integer)**

Private Sub *oggetto*_**KeyPress(**[*index* **As Integer**,]*keyascii* **As Integer)**

La sintassi dell'evento KeyPress è composta dalle seguenti parti:

Parte	Descrizione
oggetto	Espressione oggetto che definisce un oggetto dell'elenco "Si applica a".
index	Intero che identifica in modo univoco un controllo se questo è incluso in una matrice di controlli.
keyascii	Intero che restituisce un codice ANSI numerico standard. L'argomento *keyascii* viene passato per riferimento. Quando si modifica il valore dell'argomento, viene inviato all'oggetto un diverso carattere. Se si imposta *keyascii* su 0, la pressione del tasto verrà annullata e l'oggetto non riceverà alcun carattere.

Osservazioni

L'evento viene ricevuto dall'oggetto che ha lo stato attivo. Un form può ricevere lo stato attivo solo se include controlli non visibili e disattivati oppure se la proprietà **KeyPreview** è impostata su **True**. Un evento KeyPress può includere qualsiasi carattere stampabile, qualsiasi combinazione di CTRL con un tasto alfabetico o un carattere speciale, nonché il tasto INVIO o BACKSPACE. La routine dell'evento KeyPress è utile per intercettare sequenze di tasti immesse nel controllo **TextBox** o **ComboBox**, in quanto consente di verificare immediatamente la validità delle sequenze di tasti o di formattare i caratteri mentre vengono digitati. Quando si modifica il valore dell'argomento *keyascii*, verrà visualizzato un diverso carattere.

È possibile convertire l'argomento *keyascii* in un carattere utilizzando la seguente espressione:

```
Chr(KeyAscii)
```

Sarà quindi possibile eseguire operazioni su stringhe e ritradurre il carattere in un numero ANSI che il controllo potrà interpretare utilizzando la seguente espressione:

```
KeyAscii = Asc(char)
```

Utilizzare le routine di eventi KeyDown e KeyUp per gestire le sequenze di tasti non riconosciute da KeyPress, come ad esempio i tasti funzione, i tasti di modifica, i tasti di spostamento e qualsiasi combinazione di questi tasti con modificatori di tastiera. A differenza degli eventi KeyDown e KeyUp, KeyPress non indica lo stato fisico della tastiera, ma passa un

carattere.

KeyPress interpreta un carattere maiuscolo e il corrispondente carattere minuscolo come due codici distinti, ovvero come due caratteri diversi. Gli eventi KeyDown e KeyUp, invece, interpretano i caratteri maiuscoli e minuscoli in base a due argomenti, ovvero l'argomento *keycode* che indica il tasto fisico e restituisce quindi "A" maiuscolo e "a" minuscolo come lo stesso tasto, e l'argomento *shift* che indica lo stato di *shift+tasto* e restituisce quindi "A" maiuscolo o "a" minuscolo.

Se la proprietà **KeyPreview** è impostata su **True**, un form riceve questi eventi prima che li ricevano i controlli del form. Utilizzare questa proprietà per creare routine di gestione della tastiera globali.

> **Nota** Il numero ANSI corrispondente alla combinazione CTRL+@ è 0. Dato che in Visual Basic l'impostazione 0 dell'argomento *keyascii* viene considerata come stringa di lunghezza zero (""), è consigliabile non utilizzare CTRL+@ nelle applicazioni.